

Gruppenaufgabe: Effiziente Programmierung

Michael Roth

Aufgabenstellung

Ihre Aufgabe ist das Verfassen eines kleinen Programms, welches einem Benutzer Lösungsvorschläge für das Brettspiel *Scrabble* generiert.

Bei Scrabble gilt es aus einer Anzahl von Plättchen ein möglichst langes bzw. "gutes" Wort zu bilden. Jedes Plättchen enthält dabei einen Buchstaben sowie eine Punktzahl. Ein "gutes" Wort ist demnach das Wort mit der maximal möglichen Punktzahl aus allen aus den verfügbaren Plättchen bildbaren Wörter.

Hinweis: beim "echten" Scrabble müssten Sie noch die bereits auf dem Brett liegenden Plättchen mitbeachten, da Sie ja an einem bestehenden Plättchen anlegen müssen. Für diese Aufgabe allerdings betrachten wir nur die Plättchen, welche die Spielerin zur Verfügung hat.

Schreiben Sie also ein Programm welches:

1. Beim Start die weiter unten verlinkte Wortliste einliest und speichert
 - Experimentieren Sie dabei mit verschiedenen Containern (mehr dazu später)
2. Vom Benutzer (z.b. per `input`) einen String mit den verfügbaren Plättchen einlesen
 - Anschließend **alle** Permutationen von **allen** möglichen Teilmengen der Plättchenmenge bestimmen
 - Für jedes Ergebnis überprüfen, ob dieses ein deutsches Wort *gemäß der Wortliste* ergibt
 - *Hinweis:* Die unten verlinkte Liste enthält auch Abkürzungen, die normalerweise beim Scrabble nicht erlaubt wären. Für die Aufgabe ist dies jedoch irrelevant.
3. Eine nach Punkten sortierte Liste von legbaren Wörtern ausgeben.

Beispielausgabe (ohne Sortierung!) für die Plättchen `e r n s t l`:

er
ernst
erst
es
lern
lernst
lernt
lest
lrs
lt
nest
net
nl
nr
ns
nt
re
ren
rens
rest
rtl
set
st
stern
str
tel
tls

Zeitmessungen

Machen Sie Performance Messungen für verschiedene Container (`list`, `set`, evtl. `dict`) und überlegen Sie, woher evtl. auftretende Laufzeitunterschiede zustande kommen. Recherchieren und implementieren Sie auch gegebenenfalls Methoden zur Verbesserung der Performance, bspw. *binary search*. Messen Sie dabei **nur** die Zeit, die von der tatsächliche Suche der “Matches” benötigt wird, **nicht** die Zeit, die das gesamte Programm läuft. Gemeint ist hier vor allem das initiale Einlesen der Datei sowie die Benutzereingabe, welche eben **nicht** mitgemessen werden sollen.

Bereiten Sie Ihre Ergebnisse in einer Grafik auf.

Zusatzaufgaben

Es gibt (mindestens) eine Verbesserung des oben genannten Vorgehens, bei dem nicht alle Permutationen der Eingabemenge benötigt werden. Überlegen Sie sich (mindestens) eine Verbesserung und implementieren Sie diese. Vergessen Sie nicht, diese Verbesserung gegen eine bestehende Lösung zu vergleichen! Damit können Sie die Korrektheit überprüfen und, ob es überhaupt eine Verbesserung der Performance gibt.

Präsentation

Dokumentieren Sie Ihr Programm sowie Ihre Vorgehensweise und bereiten Sie diese als Gruppenpräsentation auf. Diese Präsentation sollte auch Ihre Ergebnisse bezüglich der Performance Messung enthalten.

Verweise

[Wortliste Deutsch](#)

[Übersicht der Buchstabensteine sowie der Punkte](#)